



**University of
Zurich^{UZH}**

**Zurich Open Repository and
Archive**

University of Zurich
University Library
Strickhofstrasse 39
CH-8057 Zurich
www.zora.uzh.ch

Year: 2018

A Qualitative Study on Using GuideGen to Keep Requirements and Acceptance Tests Aligned

Hotomski, Sofija ; Glinz, Martin

Abstract: Software requirements constantly change, thus impacting all other artifacts of an evolving system. In order to keep the system in a consistent state, changes in requirements should be documented and applied accordingly to all affected artifacts, including acceptance tests. In practice, however, changes in requirements are not always documented nor applied to the affected acceptance tests. This is mostly due to poor communication, lack of time or work overload, and eventually leads to project delays, unintended costs and unsatisfied customers. GuideGen is a tool-supported approach for keeping requirements and acceptance tests aligned. When a requirement is changed, GuideGen automatically generates guidance in natural language on how to modify impacted acceptance tests and communicates this information to the concerned parties. In this paper, we evaluate GuideGen in terms of its perceived usefulness for practitioners and its applicability to real software projects. The evaluation was conducted via interviews with 23 industrial practitioners from ten companies based in Europe. The results indicate that GuideGen is a useful approach that facilitates requirements change management and the communication of changes between requirements and test engineers. The participants also identified potential for improvement, in particular for using GuideGen in large projects.

DOI: <https://doi.org/10.1109/RE.2018.00-54>

Posted at the Zurich Open Repository and Archive, University of Zurich

ZORA URL: <https://doi.org/10.5167/uzh-205000>

Conference or Workshop Item

Published Version

Originally published at:

Hotomski, Sofija; Glinz, Martin (2018). A Qualitative Study on Using GuideGen to Keep Requirements and Acceptance Tests Aligned. In: 26th IEEE International Requirements Engineering Conference (RE'18), Banff, Alberta, Canada, 20 August 2018 - 24 August 2018. IEEE, 29-39.

DOI: <https://doi.org/10.1109/RE.2018.00-54>

A Qualitative Study on Using GuideGen to Keep Requirements and Acceptance Tests Aligned

Sofija Hotomski and Martin Glinz
 Department of Informatics
 University of Zurich
 Switzerland
 E-mail: {hotomski, glinz}@ifi.uzh.ch

Abstract—Software requirements constantly change, thus impacting all other artifacts of an evolving system. In order to keep the system in a consistent state, changes in requirements should be documented and applied accordingly to all affected artifacts, including acceptance tests. In practice, however, changes in requirements are not always documented nor applied to the affected acceptance tests. This is mostly due to poor communication, lack of time or work overload, and eventually leads to project delays, unintended costs and unsatisfied customers. GuideGen is a tool-supported approach for keeping requirements and acceptance tests aligned. When a requirement is changed, GuideGen automatically generates guidance in natural language on how to modify impacted acceptance tests and communicates this information to the concerned parties.

In this paper, we evaluate GuideGen in terms of its perceived usefulness for practitioners and its applicability to real software projects. The evaluation was conducted via interviews with 23 industrial practitioners from ten companies based in Europe. The results indicate that GuideGen is a useful approach that facilitates requirements change management and the communication of changes between requirements and test engineers. The participants also identified potential for improvement, in particular for using GuideGen in large projects.

I. INTRODUCTION

Changes in software requirements are inevitable and they affect all other software activities, including acceptance testing. Although requirements engineering and acceptance testing are mostly performed separately, there is much synergy between them. Requirements describe what a system should do, while acceptance tests verify that the actual behavior and properties of the system comply with the specified requirements [1]. Therefore, requirements and their corresponding acceptance tests should always be aligned [2].

In order to achieve and maintain this alignment, all changes in requirements should be documented and communicated on time to the testers so that they can change the acceptance tests accordingly. However, in practice, keeping requirements and acceptance tests aligned and up-to-date is a challenging and expensive task [3]. Therefore, automation in requirements change management [4], as well as guidelines and tool support [5] are needed.

GuideGen is a tool-supported approach that facilitates keeping requirements and acceptance tests aligned [6]. When a requirement is changed, GuideGen automatically generates guidance on how to modify the affected acceptance test. We previously evaluated GuideGen in terms of the correctness

of the generated guidance [6]. However, we did not perform a thorough qualitative evaluation of the actual usefulness of GuideGen in practice.

In this paper, we aim at closing this gap by evaluating GuideGen in terms of its usefulness to practitioners and applicability to their projects, using a qualitative experiment [7]. We presented GuideGen to practitioners using their own data and then conducted thirteen interviews in ten companies with a total of twenty-three experienced practitioners. In order to obtain a baseline for comparison, we started the interviews with an exploratory part where we investigated the current challenges that the participants are faced with when requirements change. Then we assessed the usefulness of GuideGen in practice by asking the participants which of their current challenges can be mitigated by using GuideGen and how applicable and useful GuideGen would be in their current projects.

The results show that the challenges about aligning requirements and acceptance tests found in earlier studies (e.g., [3], [8], [9]) are still present, indicating that new approaches are needed for managing changes in requirements and keeping them aligned to acceptance tests. The principal idea of GuideGen was rated to be useful or even extremely useful by all interviewed practitioners, and even in the practical setting of the current projects, 20 of the 23 participants found GuideGen to be useful or extremely useful. The participants also identified things to be changed or added to the GuideGen tool in order to make it fully applicable in practice.

The remainder of the paper is structured as follows. We first give a short overview of the GuideGen method and tool in Section II. In Section III, we describe our research methodology. In Section IV we present the results and discuss them in Section V. Related work is discussed in Section VI. Section VII concludes the paper.

II. GUIDEGEN

GuideGen¹ is a tool-supported approach for automatically generating guidance on how to update acceptance tests when their corresponding requirements change [6], [10]. To make this paper more self-contained, we briefly describe the GuideGen method and tool with a running example from the healthcare domain. Figure 1 shows the list of requirements for a patient management system in the GuideGen tool.

¹<http://www.guidegen.org>

Requirement ID	Requirement Title	Related Test
REQ_1	Add a new patient	TEST-1
REQ_2	Remove a patient	TEST-2
REQ_3	Make a new appointment	TEST-3
REQ_4	Cancel an appointment	TEST-4

Fig. 1. A list of requirements and their acceptance tests

We assume that a requirements engineer needs to change requirement REQ4 about cancelling an appointment. She clicks on the requirement ID in the tool and gets a window where she can edit the requirement as follows (added parts are in green and underlined):

“A patient can cancel his/her appointment. The doctor’s office and the patient will get a cancellation e-mail. The patient must be logged-in first in order to make the cancellation.”

As soon as the changes are applied to the requirement, GuideGen analyzes the changes and determines the relevant parts, i.e., those that impact an acceptance test linked to the changed requirement. In our example, the addition of the subject “patient” and of a complete sentence are relevant, while the added words “and” and “the” are irrelevant. Based on this analysis, GuideGen automatically generates guidance in natural language for the testers about how to change the associated acceptance test. The guidance consists of one suggestion per relevant changed element. In our example, the following suggestions are generated:

1. Make sure that now the patient will get a cancellation e-mail. Add the steps or modify the expected results which verify this activity.
2. Add new steps or modify existing steps to verify that the patient must be logged-in first in order to make the cancellation.

The details about how the analysis works and the suggestions are generated can be found in [6].

The GuideGen tool displays the generated guidance to the requirements engineer (Fig. 2) who can now review the generated suggestions, mark those that she considers to be irrelevant or wrong and then let GuideGen notify all people who have subscribed to such notifications.

Notification works by an automatically generated e-mail, which contains the information about the affected acceptance test, the relevant suggestions on how to adapt that test and the summarized changes in natural language.

In the list of requirements, GuideGen also flags the acceptance test associated to the changed requirement with a warning sign, indicating that this test is outdated. When a test engineer receives a notification message from GuideGen, he can open the affected acceptance test case and edit it. GuideGen displays the editing window with the text of the test and the suggestions about what to change side by side (Fig. 3). When the tester saves his changes, GuideGen removes

Change the acceptance test:

1. Make sure that now the patient will get a cancellation e-mail. Add the steps or modify the expected results which verify this activity.

Ignore

2. Add new steps or modify existing steps which verify that the patient must be logged-in first in order to make the cancellation.

Ignore

Cancel Email subscribers

Fig. 2. A list of suggestions generated by GuideGen

the warning flag from the list of requirements. More details about the GuideGen tool can be found in [10].

Edit test case

Description
Test for canceling an appointment

Steps and Expected Results
Step 1. Go to the list of the appointments and click on the label Cancel next to the appointment
Expected: A pop-up window is shown

Step 2. Click on the Cancel button to confirm the cancellation.
Expected: The appointment is cancelled, removed from the list of the appointments and an email is sent

1. Make sure that now the patient will get a cancellation e-mail. Add the steps or modify the expected results which verify this activity

2. Add new steps or modify existing steps which verify that the patient must be logged-in first in order to make the cancellation

Cancel Save

Fig. 3. Suggestions that are shown to a test engineer in the affected test case

III. RESEARCH METHODOLOGY

A. Research Goal and research questions

In [6] we have evaluated GuideGen in terms of the correctness and appropriateness of the guidance generated by GuideGen. With a correctness score of more than 80% for real-world agile requirements and around 67% for traditional requirements, we found that GuideGen provides relevant and useful guidance for keeping acceptance tests aligned with their requirements. However, we did not assess the actual usefulness of GuideGen in practice. In this paper, our goal is to close this gap by performing a qualitative evaluation of GuideGen in terms of its perceived usefulness for practitioners and its applicability to industrial software projects. From this goal, we derive four research questions:

- RQ1: What are the current challenges with evolving requirements faced in industry?
- RQ2: What are the benefits of GuideGen that can have a positive impact on the current challenges?
- RQ3: How applicable is GuideGen to the current setup of real industrial projects?
- RQ4: How useful is GuideGen for solving the currently faced issues with evolving requirements?

With RQ1, we assess the current situation with documenting and communicating changes in requirements and their tests

and thus establish a *baseline*, against which the usefulness of GuideGen can be assessed. RQ2 investigates the benefits of GuideGen in order to assess its potential usefulness as perceived by industrial practitioners. RQ3 studies how well GuideGen can be used in the framework of the processes that practitioners currently use in their projects. Finally, RQ4 sheds light on the concrete usefulness of GuideGen for solving issues that practitioners are faced with in their current projects.

B. Study Design

To answer our four research questions, we performed a *qualitative experiment*², which consists of three parts.

1. *Introductory presentation*. The first author explained the GuideGen approach in a 20 minute talk to a group of business and technical practitioners. Then, a five minute video³, which demonstrates the GuideGen tool, was shown to the audience, followed by a question and answer session of up to 30 minutes about both the GuideGen approach in general and the GuideGen tool. The first author then asked for requirements and test engineers who were interested to try out the tool and to evaluate GuideGen. We did this to ensure that only people who are interested and knowledgeable in the field of requirements engineering or testing participated in the experiment.

2. *Tool trial*. Although the participants already got familiar with the GuideGen tool by attending the presentation, we encouraged them to try it themselves, by using examples from their own projects. The tool trial lasted between 30 and 45 minutes.

3. *Interview*. After a five minute break, we conducted semi-structured interviews [11] with the participants. Each interview consisted of two main parts: *an exploratory part* referring to the current issues and challenges faced when requirements change (RQ1) and *an evaluation part* about the usability and applicability of GuideGen (RQ2-RQ4). The interviews lasted between sixty and ninety minutes.

When there were more than two participants available or the presentation was held after the working hours, the first author came back the day after, otherwise the tool trial and interviews were held right after the presentation. The first author visited seven of the ten companies in person. For three companies (C6, C7 and C9 in Table I) the presentation was held via Skype and only to the participants who were actually interviewed.

1) *Demographics*: The study is based on 13 interviews with 23 participants from ten companies. The companies are distributed over five countries in Europe: three companies are located in Switzerland, four in Serbia, one in Germany, one in Denmark and one in the Netherlands. We selected the companies by convenience sampling, using our personal contacts. We only selected companies that have documented requirements and acceptance tests, written in natural language. An overview of the companies is shown in Table I.

²According to Kitchenham [7], a qualitative experiment is “a feature-based evaluation done by a group of potential users who are expected to try out the methods/tools on typical tasks before making their evaluation”.

³The video is available at <https://tinyurl.com/y9leos5e>

TABLE I
AN OVERVIEW OF THE COMPANIES

Company	Domain of activities	Country	# of employees
C1	Power industry	Serbia (International)	1000 (10000 worldwide)
C2	Access-control solutions	Switzerland (International)	1000 (10000 worldwide)
C3	Distribution Solutions	Switzerland (International)	500 (2500 worldwide)
C4	Enterprise Content Management	Switzerland (International)	50 (7000 worldwide)
C5	IT project and product management	Serbia (Germany, India)	20 (500 in total)
C6	Online marketing and online shopping	Germany (International)	1200 (3000 worldwide)
C7	Content Management Systems	Netherlands (International)	60 (600 worldwide)
C8	Software testing services	Serbia (Austria, Switzerland)	120 (500 in total)
C9	Data and Analytics	Denmark (Sweden)	250 (1000 in total)
C10	Website and mobile development	Serbia (Germany)	110 (350 in total)

All participants have between five and seventeen years of experience in IT and at least three years of experience in the requirements engineering and/or testing field. An overview of the participants and their teams is provided in Table II. In total, we interviewed twelve test and quality assurance engineers, five requirements engineers and six industrial practitioners who are doing both requirements engineering and acceptance testing. We also present the size of the teams and the software process model applied within the teams, since that may impact the way that changes are communicated and documented, indirectly influencing the applicability of the approach.

2) *Data collection and analysis*: All experiments were conducted in December 2017 by the first author. The majority of experiments were face-to-face, except for companies C6, C7 and C9 that were conducted via Skype, due to their location. When the presentation and the trial of the tool were not done in person but via Skype, we shared the screen and the participants dictated or copied in Skype chat the text that should be inserted into the tool. In such a way the participants P16, P17 and P21 could see how the tool performs using their own data, just as other participants who tried the tool.

The interview instrument⁴ was prepared in advance and has been discussed with several researchers specialized in qualitative studies and/or requirements engineering. In addition, we conducted a pilot interview with one personal contact from industry before starting the interviews reported in this study. Six interviews were done in pairs, two were conducted with three people at the same time and the remaining five were

⁴<https://tinyurl.com/y9hfgwn7>

TABLE II
AN OVERVIEW OF THE PARTICIPANTS AND THEIR TEAMS.

Participant/ Company	Interview ID	Role	Years in IT / Years in RE or QA field	Team structure	Team size	Process model
P1/C1	I1	Test manager (QA)	8 / 8	cross-functional, co-located	10	Agile (Scrum)
P2/C1	I1	Test manager (QA)	5 / 5	cross-functional, co-located	15	Agile (Scrum)
P3/C1	I2	Subject matter expert (RE)	10 / 3	cross-functional, co-located	15	Traditional, Agile (Scrum)
P4/C1	I2	Test lead (QA)	5 / 5	cross-functional, distributed*	5	Traditional, Agile (Scrum)
P5/C1	I2	Subject matter expert (RE)	10 / 4	cross-functional, co-located	15	Traditional, Agile (Scrum)
P6/C1	I3	Senior test engineer (QA)	10 / 5	cross-functional, distributed*	30	Traditional, Agile (Scrum)
P7/C1	I3	Senior test engineer (QA)	12 / 5	cross-functional, distributed*	30	Traditional, Agile (Scrum)
P8/C1	I4	QA team lead (QA)	7 / 7	cross-functional, distributed*	20	Traditional
P9/C1	I4	Senior test engineer (QA)	5 / 5	cross-functional, distributed*	20	Traditional
P10/C2	I5	Product owner (RE)	10 / 10	mixed, co-located	8	Agile (Scrum)
P11/C2	I5	Senior test analyst (QA)	12 / 12	mixed, co-located	8	Agile (Scrum)
P12/C3	I6	QA manager (QA)	12 / 11	mixed, distributed	10	Traditional
P13/C4	I7	Head of operation (RE, QA)	17 / 17	mixed, distributed	1-5	Traditional
P14/C4	I7	Consultant (RE, QA)	17 / 17	mixed, distributed	10	Traditional
P15/C5	I8	Product manager (RE, QA)	6 / 4	mixed, distributed	4-5	Agile (Scrum)
P16/C6	I9	Senior Agile coach (RE, QA)	12 / 10	mixed, co-located	10-20	Agile(Scrum)
P17/C7	I10	Solution consultant (RE)	10 / 4	mixed, distributed	10	Agile (Scrum, Kanban)
P18/C8	I11	Agile coach (RE, QA)	12 / 10	mixed, co-located	3-9	Agile (Scrum)
P19/C8	I11	Test manager (QA)	5 / 5	cross-functional, distributed	10	Agile (Scrum)
P20/C8	I11	Test manager (QA)	6 / 6	cross-functional, distributed	14	Agile (Scrum)
P21/C9	I12	Solution architect (RE)	6 / 4	mixed, distributed	15	Agile (Scrum)
P22/C10	I13	Software developer (QA)	6 / 2	mixed, distributed	8-10	Agile (Scrum)
P23/C10	I13	Team lead (RE, QA)	6 / 2	mixed, co-located	10	Agile (Scrum)

*located in different buildings within the same city.

RE - requirements engineers, QA - test and quality assurance engineers.

done with one person, as shown in Table II. All interviews were conducted in English.

The interview instrument contains six parts, with questions referring to: (1) the interviewee's profile, (2) the profile of the company, (3) the software process model applied within a team, (4) how requirements and acceptance tests are documented and linked, (5) how requirements and acceptance tests are updated, and (6) the evaluation of GuideGen in terms of its usability and applicability. When asked about how requirements and acceptance tests are managed, documented and updated (parts 3, 4 and 5), the interviewees explained also the issues they are faced with, thus answering RQ1. In part 6, we asked how GuideGen could influence the communication (6.5) and the way of working with requirements and acceptance tests (6.4), thus learning about the perceived benefits of GuideGen (RQ2). The applicability of GuideGen (RQ3) was assessed by asking whether the current implementation of GuideGen fits to the interviewees' current projects (6.6-6.9). Finally, we learned about the usefulness of GuideGen by directly asking the interviewees for their opinion (6.1, 6.2 and 6.8).

For analyzing the data, all interviews were recorded and later transcribed. The answers from all participants for each of the questions were grouped in order to be able to compare them. In such a way it was possible to quantify the answers to certain topics during the analysis.

IV. RESULTS

In this section, we present the results of our study, grouped by our four research questions.

A. Current issues with changing requirements (RQ1)

We first investigated the current challenges with changing requirements faced by the industrial practitioners. Although previous research [3], [8], [9] already has identified the most common challenges, we needed to assess the current state of practice as seen by the participants of our study, in order to set up a baseline for the subsequent evaluation of the usefulness and applicability of GuideGen. The results show that the challenges identified in previous work are also perceived by the participants of our study. The most frequently mentioned challenges concern communication, documentation and tools.

Communication issues. When asking about the difficulties faced when requirements evolve, communication issues were explicitly mentioned by ten participants (P1, P4, P6, P8, P10, P12, P14, P16, P21 and P22) and agreed by seven other participants who were interviewed together with one of those ten (P2, P3 and P5, P7, P9, P13 and P23). Not surprisingly, all seventeen participants work in distributed or cross-functional teams, except P16 who explained that the communication problems occur when: *"people are coming and leaving the company, they don't know all the communication channels that we use in the company or they don't know the whole domain"*.

The most frequently reported communication issues are: late- or non-communicated changes to test engineers (CI1)⁵, passing incomplete information from requirements to test engineers (CI2) and inability to track changes due to different communication channels (CI3).

⁵We assign identifiers to all issues so that we can refer to them later.

Several researchers and practitioners advocate the involvement of test engineers in all requirements related activities as a best practice [2], [3], [12]. However, in many companies this practice is still not applied. P1, P4, P6, P12 and P14 stated that changes in requirements are not communicated on-time to test engineers due to excluding testers from requirements activities. P2 framed the problem as follows: *“They [the requirements engineers and the top management] don’t understand how important it is and we don’t have that mindset that product owners and test engineers are “brothers”. They need to work together and write documents also together, to be sure that they are on the same page”*.

Moreover, P21 and P22 said that sometimes changes in requirements are not communicated to testers at all: under time pressure, requirements engineers often communicate changes only to developers.

Another problem, mentioned by P6, P8 and P12, is passing incomplete information from requirements engineers to test engineers. This causes test engineers to apply incorrect changes to test scenarios. Test managers P8 and P9 mentioned that the quality of communicated information sometimes depends on the mood of requirements engineers. P8 explained: *“Sometimes it goes smoothly and we clarify everything, but sometimes if they are in a bad mood we either don’t hear from them at all, or we have unpleasant conversations, we don’t get enough data and we cannot later do our job correctly”*.

Further, P4 and P16 reported difficulties for test and requirements engineers to track changes when different communication channels are used at the same time, such as verbal communication, e-mails and chats.

Documentation issues are also present. The most frequently mentioned ones are: outdated documents (DI1), incorrect acceptance tests (DI2) and bad quality of requirements documents (DI3).

While there is research addressing the problem of outdated documents (e.g., [13], [14]), this is still a major issue in practice. P15 stated: *“The documentation is the first thing to suffer if there is no time or [when there is] a lot of stress or a lot of people involved”*. P7 explained that this is due to lack of time and a pressure from clients. The problem of outdated documents is also mentioned by P2, P8 and P12.

Incorrect acceptance tests are another issue that occurs mostly when testers are new and do not understand the context or when they are excluded from designing or modifying requirements, explained P5, P6, P11, P16 and P17. Another reason for having incorrect tests is that test engineers mostly rely on their own experience when writing or modifying tests. This is due to bad quality of requirements documents and poor communication between test and requirements engineers.

The most common issues with regard to the bad quality of requirements documents are: incompleteness or lack of context (P2, P6, P9, P10, P16, P17, P22), outdated requirements (P1, P3, P7, P8 P15, P12) and lack of traceability links between requirements and tests (P8, P14). Besides the lack of time, another reason for outdated, incomplete and non-traceable requirements documents are lack of knowledge with regard

TABLE III
TOOLS CURRENTLY USED FOR MANAGING REQUIREMENTS AND
ACCEPTANCE TESTS DOCUMENTS

Tool	Document type	Number of participants	Participants’ ID
TFS	Low-level requirement	11	P1-P9, P13, P14
	Acceptance test	10	P1-P9, P13
	Acceptance criteria	2	P1, P2
Jira	Low-level requirement	10	P10, P11, P15-P21, P23
	Acceptance criteria	3	P18, P21, P23
Polarion	Low-level requirement	2	P12, P22
	Acceptance test	2	P12, P22
Microsoft Word	High-level requirement (functional spec.)	10	P3-P9 P13, P14, P17
Excel	High-level requirement	7	P3-P9
	Acceptance test	1	P14
Confluence	High-level requirement	2	P13, P14
TestLink	Acceptance test	9	P3-P9, P16, P15
QAComplete	Acceptance test	2	P10, P11
iScrum	Acceptance test	2	P1, P2
Google Docs	Acceptance test	2	P15, P16
Tosca	Acceptance test	2	P19, P20
Test rail	Acceptance test	1	P17

to change impact and risk (P1, P16, P17) and preference for verbal communication (P10, P11).

Issues with the currently used tools were also reported frequently: usage of too many different tools for document management (TI1), complexity of the tools (TI2) and inconvenient notification system (TI3).

Table III shows the reported tools used for requirements and test document management. The most common tools for managing low-level or technical requirements are TFS and Jira. P12 and P22 said that they use Polarion for this purpose. For high-level requirements practitioners use Microsoft Word, Microsoft Excel or Confluence. The tools for managing manual acceptance tests are TFS, TestLink, Jira, QA complete, iScrum, Google Docs, Tosca, Polarion, Excel and Test rail.

Using too many different tools and having requirements and tests in various places is reported to be challenging (P2, P5, P16). For instance, P5 explained that they first get high-level requirements from their clients in an Excel document and then, based on that document and discussions with the clients, they create functional specification in Word, as an internal, high-level requirements document. From the functional specification they derive user stories for development, which is their technical requirement specification. P5 stated: *“Sometimes you really need to be experienced to quickly find information that is needed. You need to know what is the level of detail in order to know where to look for the information”*.

Another problem is the complexity of the current tools. For instance, in TFS or TestLink too much “meta-work” is needed to get basic information, such as a traceability link or a history of changes, as reported by P2 and P15. P15 stated: *“Too many clicks are needed and meta-steps in order to create a test case or to see the link between a test and its requirement. This is why I use Google Docs for smaller projects, but I am aware that this is not sustainable”*.

The most common problem reported with regard to the notification system in TFS is that a user must subscribe for every feature individually and personally, and there is no possibility that a user subscribes other users to a feature of interest. When there are too many features, practitioners sometimes miss to subscribe and are not notified about important changes.

Another issue with the notification system in the currently used tools is spamming, as reported by P6, P15, P22 and P23. In case of numerous changes practitioners get too many notifications and, therefore, some changes are missed or simply ignored.

B. Benefits and Strengths of GuideGen (RQ2)

After discussing the current challenges, we assessed which of them GuideGen can mitigate. We asked the participants how using GuideGen would influence their work. From their answers, we identified seven benefits of GuideGen (labeled B1-B6) which we present below.

B1. Faster communication of changes. Participants who work in cross-functional or distributed teams saw a benefit in the mechanism for communicating changes provided by GuideGen. For instance, P20 reported: *“It can also shorten time for the communication of changes and just to keep test documents aligned”*. Moreover, this benefit is recognized also by the participants who work in co-located environment, such as P1 and P10. For instance, P10 explained: *“In the companies where people don’t sit next to another the tool would help to communicate changes more quickly”*. This benefit directly addresses the issue with late communication of changes (CI1) and indirectly the issue with writing incorrect acceptance tests (DI2).

B2. Changes no longer missed or overseen by testers. Several participants stated that with GuideGen it would happen much less that changes are communicated badly or not on-time. P7 explained: *“A lot of things would not pass by us. I would get the information about every change. That would make the process of maintaining the change and documents easier”*. P11 also saw the following benefit: *“If I am away for a business trip or on holidays, we would not miss (...) changes [while being absent]”*. This again refers to CI1 and to passing incomplete information from requirements to test engineers (CI2).

B3. Acceptance tests updated faster and easier. Participants stated that suggestions, automatically generated e-mails and warnings would result in faster and more timely updates of acceptance tests. For instance, P21 explained: *“It would be definitely faster and smoother to communicate changes and to apply the changes, because we would not need to think that*

much what to do, if the system would propose the changes based on the updates in the requirement. It helps that the whole process goes faster”. The participants P1, P3, P4, P5, P8, P13, P15 and P20 also made similar statements.

Updating acceptance tests would not only become faster with GuideGen, but also easier. As P2 explained: *“It will be much easier, because we would have alerts for the changes and we would really use it to know how to change test cases and not only that the change is there”*. Participants P6, P7, P20 and P23 made similar statements.

By improving the process of communicating and understanding changes (B3), GuideGen addresses the issues CI1, CI2 and CI3. If industrial practitioners apply the changes suggested by the tool, that would positively affect the issue with outdated documents (DI1).

B4. Requirements and acceptance tests kept closely together. In some companies, requirements and acceptance tests are stored and managed in different tools, thus keeping them and their changes easily traceable becomes challenging. For instance, P5 explained: *“If I change something in a requirement, currently testers don’t have a clue that something is changed. I think it’s good to keep requirements and acceptance tests together and that changes are visible to everybody”*. This benefit directly addresses the issue with using too many tools for requirements and test management (TI1) and indirectly the issue with the quality of requirements (DI3), since GuideGen establishes traceability links between requirements and tests. P15 and P19 made similar statements.

B5. Simple to use. The GuideGen tool misses many features that other tools have (see IV-C2). P15 and P23 perceived this as a benefit and stated that due to its simplicity they would use GuideGen for smaller projects rather than TFS or Jira. P15 explained that for smaller projects she uses Google docs in order to avoid the complexity of TFS and Jira and that GuideGen provides much better possibilities for document management than Google docs. This partially addresses the issue with the complexity of the current tools (TI2).

B6. Serving as a reminder for updating. P18 rated the way of communicating changes with GuideGen as follows: *“It would be a good reminder and a good summary.”* P12 rated the warnings as follows: *“I see it as a good reminder for me not to forget to update a test case”*, while P11 emphasized: *“E-mails can easily be ignored, but warnings in the app itself will remind us that the documents are non-aligned and that there has been a change in the requirement. The warning cannot be ignored forever”*. This partially solves the issues DI1 (outdated documents), because test cases will not stay outdated, and TI3 (inconvenient notification system).

C. Applicability of GuideGen to real industrial projects (RQ3)

In order to assess how applicable GuideGen is to the current setup in the companies, we specifically asked the participants if they would use the GuideGen tool for their projects if it were available. Two participants said that they would use it for smaller projects as is (P15, P23), while 14 (P5, P9-P14, P16-P22) said that they would use GuideGen if it were

available as a plug-in for their existing tools. The remaining 7 participants (P1-P4, P6, P7, P8) said they would use it, but would need more features. Two of them (P6 and P7) mentioned that GuideGen could be a standalone application, but then the changes should be automatically replicated to the currently used tools, such as Jira or TFS.

In the following two sub-sections we present the major concerns towards the applicability of GuideGen and what features the participants missed in the GuideGen tool.

1) Major concerns about the applicability of GuideGen:

The majority of participants was concerned about the data migration effort when switching to a new tool, in particular, when the amount of documents is high.

Next, there are different types of requirements and acceptance tests, defined with different levels of detail, while GuideGen currently supports only one type of requirements and tests. In real projects, requirements can be specified as high level requirements, followed by more detailed functional specifications, epics, user stories, etc. Acceptance tests can be, for instance, internal acceptance tests which usually correspond to user stories or business acceptance tests which usually consist of a couple of functional tests, covering one business unit. Furthermore, some companies do not have acceptance tests, but only acceptance criteria defined within user stories, explained P18, P20 and P23.

Further, participants were concerned about the bad quality of requirements in their company and how that may influence the performance of the GuideGen tool. Too short and incomplete sentences, too long sentences and grammatically incorrect sentences are often present in requirements. Changes in such sentences might cause that the tool generates incorrect suggestions. Another concern are outdated requirements that directly hinder the applicability of GuideGen.

Finally, the problem with spamming by sending too many e-mails due to frequent changes was another concern.

2) *Features missing in GuideGen:* We then specifically asked about the features that would be needed to make GuideGen applicable to the participants' current projects, but are missing in the current version of the GuideGen tool. The missing features mentioned in at least two interviews are presented in Table IV. They are mostly related to the workflow and status management or the usability.

Other features mentioned by at least one participant are: an improvement of the user interface for documenting test cases (P4), linking test cases to a specific acceptance criterion and not to the whole requirement or user story (P1), multiple languages support (P9), identifying which test case is actually impacted among many related ones (P15), identifying which concrete step in the test case is affected (P11), a possibility to make a test plan and not only scenarios (P2), including bugs in the system besides requirements and test cases (P2), ability to change and adapt the generated suggestion (P5), an automated analysis of changes in diagrams and images (P8), proposing manual checking instead of generating suggestions in case when a requirement is changed more than 60% (P8), a possibility to send a message to all subscribers on Slack

TABLE IV
THE MISSING FEATURES REPORTED BY AT LEAST TWO PARTICIPANTS IN TWO DIFFERENT INTERVIEWS

Missing features	Participants
A possibility to see the outcome of the test execution (e.g. skipped, passed, failed, not started...)	P2, P4, P6, P12, P15, P19, P21
A possibility to see the status of a requirement (e.g. new, assigned, opened, finished...)	P6, P12, P13, P15, P17, P21
Tracking the history of changes in requirements and tests	P5, P7, P16, P20, P23
Personalize the user (having different accounts with different privileges)	P7, P9, P14, P16, P21
Different structure for keeping the documents (more folders for grouping different types of requirements, different projects or features)	P3, P5, P6, P12

instead of sending e-mails (P16), a central place to maintain boards like in Jira and estimation and prioritization tools (P17).

The current version of the GuideGen tool is limited to a one-to-one relationship between requirements and acceptance tests [10]. To assess the severity of this limitation, we explicitly asked about what relationship is the most common in practice. Two participants stated that they mostly have 1:N, another two reported 1:1, one participant reported M:1 and the remaining participants said that they have an M:N relationship. Two of those who reported M:N explained that due to lack of time they often have 1:1, but they aim at having multiple tests per requirement in the future.

D. Quantitative evaluation of usefulness (RQ4)

We performed a short quantitative assessment of GuideGen by asking the participants to evaluate its usefulness on a five point scale, from 1 (totally useless) to 5 (extremely useful). We asked about the usefulness of GuideGen both in general and for the participants' companies. The summary of the results is shown in Figure 4.

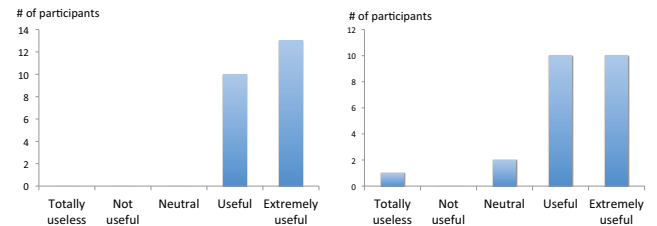


Fig. 4. The usefulness of GuideGen in general (left) and for the company (right)

The participants P1-P5, P8-P11, P14, P16, P17 and P23 evaluated the approach as extremely useful and P6, P7, P12, P13, P15, P18-P22 as useful in general. When we asked for their work in their current company, the participants P1, P3-P5, P7-P9, P14, P15 and P23 rated GuideGen as extremely useful, P2, P6, P10-P13, P17, P19, P21, P22 as useful, P18 and P20 were neutral and P16 said that as a standalone tool

it would be totally useless for his work. P18 explained that he is neutral due to not enough time to rate the usefulness of GuideGen thoroughly. P20 said that in his team acceptance tests are specified and updated by the clients and, therefore, GuideGen would not affect his work. P16 explained that GuideGen is useless for his work as a standalone tool due to challenges faced when switching to yet another tool. However, he said he would rate GuideGen as extremely useful if it were implemented as a plug-in for Jira.

We then asked our participants to rate the usefulness of the suggestion feature (i.e., the suggestions generated by GuideGen about how to change an affected acceptance test). We gave them two scenarios, (1) the generated suggestions are correct in 98% of all cases and (2) they are correct in 80% of the cases, and asked them to rate the usefulness on a scale from 1 (totally useless) to 5 (extremely useful) for both scenarios. The results are shown in Figure 5.

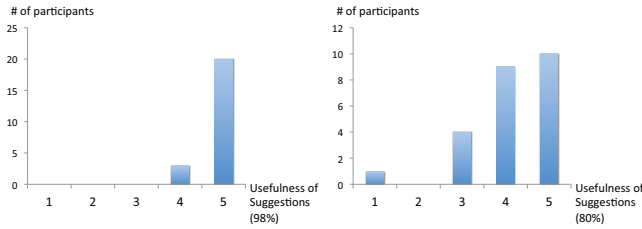


Fig. 5. The usefulness of the suggestions when they are 98% correct (left) and 80% correct (right). 1 - totally useless, 2 - useless, 3 - neutral, 4 - useful, 5 - extremely useful

In scenario 1 where 98% of the generated suggestions are correct, all participants rated this feature as useful or extremely useful. In scenario 2 with 80% correct suggestions, only one participant (P19) said that suggestions would be useless at this level of correctness. Four participants (P5, P12, P18 and P21) gave a neutral rating and explained that it might be confusing to get 20% wrong suggestions. The other 19 participants rated the suggestion feature to be useful or extremely useful even at a correctness level of only 80%.

In addition, we asked whether suggestions are needed or the summary of the changes would be enough. 18 participants explicitly stated that they need suggestions, while the others stated that it might be enough to have only the summary, but having the suggestion feature would nevertheless be useful and “nice to have”.

E. Threats to Validity

Every qualitative research is exposed to several threats to validity. In this subsection we describe the possible threats.

Construct validity This aspect of validity reflects to what extent the operational measures that are studied really represent what is investigated [15]. We are aware that the participants would need to use GuideGen for a longer period of time in order to provide a more rational evaluation of its usefulness. However, due to the limited availability of the participants and the complexity of their systems and documentation, this

was not possible. Therefore, we designed our study in such a way that the participants first got familiar with GuideGen through the presentation and the video. Moreover, after that introduction they had the opportunity to try out the GuideGen tool themselves, using examples from their own work.

The interviews were conducted by the researcher who created GuideGen, which might cause politeness bias in the interviewees’ answers. We addressed this threat by telling the interviewees that they should be objective and express all their doubts and concerns about GuideGen.

Another threat to construct validity are misunderstandings between interviewer and interviewees. In order to mitigate this threat, we tested our interview instrument in a pilot interview. In the interviews, we clarified questions whenever we had the impression that an interviewee had misunderstood them. The interviewees were also free to ask questions at any time.

Internal validity refers to credibility of drawn conclusions in a study [16]. In order to limit threats to internal validity, a good research design is always of crucial importance [17]. Therefore, we carefully designed our interview questions, we discussed them with a group of researchers, improved them and performed a pilot interview with a requirements engineer, who is our personal contact in industry. The discussion with researchers regarding the interview design helped us to limit also the reliability threat. However, reliability threats regarding researcher bias cannot be completely ruled out, because the interviews were conducted by only one researcher.

In order to avoid selection bias, we avoided personal contacts when we performed the interviews for the study. In addition, we interviewed only industrial practitioners who were interested to try GuideGen and answer our questions.

External validity is concerned with the generalizability of the findings of a study [15]. In order to limit the threat to external validity, we tried to keep diversity in terms of structure of the chosen companies, their size, domain of responsibilities, processes applied within the companies and the way that requirements and tests are documented. In addition, we chose participants based on their roles, so that we discuss with the target group, which are practitioners responsible for requirements and test documentation. Furthermore, we tried to achieve some geographical diversity.

V. DISCUSSION

In this section we discuss the study results and the lessons learned.

A. Current issues with changing requirements and how GuideGen can mitigate them (RQ1, RQ2)

Table V summarizes the most frequently reported issues and the benefits of GuideGen that can mitigate those issues. We found that late communication, miscommunication and even non-communication of changes as well as outdated documentation are still major issues in practice, which confirms the results of a previous study [3].

Some issues are mitigated by GuideGen in its current form. For example, e-mails and warnings created by GuideGen support a better change communication and ensure that changes

TABLE V
THE REPORTED ISSUES AND BENEFITS OF GUIDEGEN THAT MITIGATE THEM.

Issue type	Issue description (ID)	Benefit ID
Communication	late-/non-communicated changes (CI1)	B1, B2, B3
	Passing incomplete information (CI2)	B2, B3
	Many communication channels (CI3)	B3
Documentation	Outdated documents (DI1)	B3, B6
	Incorrect test cases (DI2)	B1
	Bad quality of requirements (DI3)	B4
Tools	Too many different tools (TI1)	B4
	Complexity (TI2)	B5
	Inconvenient notification system (TI3)	B6

are not overseen. However, in projects with a high change rate, the use of GuideGen might result in so many e-mails and warnings that the testers would feel spammed. Addressing this issue requires adaptations in GuideGen: the GuideGen notification system would have to be made configurable so that, for example, testers could opt for receiving only one message per day from GuideGen with a summary of all changes (as suggested by P15). The tool should also permit a user to subscribe other interested parties to a feature, as this is one of the major issues with the notification system of currently used tools.

There are also issues, particularly those concerning organizational culture, which cannot be solely addressed with a tool such as GuideGen. While GuideGen facilitates the process of keeping acceptance tests up-to-date (by providing guidance and reminders for test engineers), GuideGen cannot force test engineers to apply the changes. Consequently, GuideGen cannot solve the issue of outdated acceptance tests if test engineers do not update them.

The fact that GuideGen only has a small amount of features can be seen as both a weakness and an advantage. The weakness is that GuideGen cannot support projects having, for example, different levels of requirements and tests with many-to-many relationships between them. GuideGen also does not provide any sophisticated means for managing requirements and tests or for tracking issues. The advantage is that GuideGen is easy to use and suffices for smaller projects where tools like Jira or TFS are reported to be too complex and overwhelming for the users.

B. Applicability of GuideGen (RQ3)

In terms of applicability, we found that (1) the GuideGen tool can be used as is for smaller projects, but cannot replace the existing tools used for complex projects, and (2) it is most applicable to low level requirements and acceptance test documents.

1) *GuideGen as a standalone tool*: GuideGen cannot replace currently used tools for three main reasons: (1) missing features, (2) the complexity of data migration and configuration setup, and (3) the reluctance of industrial practitioners to learn new tools and adjust their working procedures accordingly. The participants stated that GuideGen needs to provide more features, for example, for managing workflow, history of changes, and access rights, as well as a possibility to better organize documents. Next, in complex systems data migration is a difficult and expensive task. The participants P16 and P17 explained that they do not want to start using another tool when they already have all data and configurations set up in the currently used tools. Finally, participants argued that learning how to use a new tool and adjusting to it also takes some time.

However, all participants see a lot of benefit in the suggestions created by GuideGen and in the proposed way of communicating changes. Therefore, many of them suggested that the tool should be a plug-in for the tools they already use. So they could profit from the core features of GuideGen without having to migrate their data or to adjust their processes to a new tool set.

2) *Types of documents to which GuideGen is applicable*:

There are different types of requirements and acceptance test documents, written with different levels of detail, as explained in Section IV-C1. GuideGen is most applicable to low-level requirements and functional acceptance tests, since their levels of detail match and changes in one document directly cause changes in another one. GuideGen is not applicable to comprehensive requirements documents whose level of detail and scope go way beyond the specification of functional and non-functional requirements, but also contain architecture details or process descriptions. Changes of those elements may not impact changes in acceptance tests.

GuideGen has been built for supporting manual acceptance testing. With a minor adaptation of suggestions and an integration within the existing frameworks for automated testing, the approach could also be applied to scenarios written using Gherkin syntax [18], which is widely used for specifying automated acceptance tests. We have not explored generating suggestions for how to change the test code when using automated, executable tests.

Since GuideGen analyzes textual requirements by using NLP methods, it is applicable only to textual documents. GuideGen cannot analyze models, graphs or images.

C. Usefulness of GuideGen (RQ4)

Davis [19] defines perceived usefulness⁶ and perceived ease of use⁷ as the fundamental determiners of user acceptance. In this study, we concentrated on perceived usefulness. Ease of use was addressed indirectly with question 6.8 in the interview

⁶The degree to which a person believes that using a particular system would enhance his/her performance.

⁷The degree to which a person believes that using a particular system would be free of effort.

instrument. Benefit B5 (see Sect. IV-B) indicates that GuideGen is easy to use. The results show that most participants in our study perceived GuideGen as useful or extremely useful. However, some participants were less positive when they evaluated the usefulness of GuideGen for their own current work. This is, for instance, due to non-frequent requirements updates in their companies, as well as their reluctance to migrate all their data into a new tool or to learn yet another tool.

The usefulness of GuideGen could be improved by implementing GuideGen as a plug-in for currently used tools in companies or by implementing missing features. With regard to the usefulness of the notification system, the e-mail generation mechanism in GuideGen should be adapted in order to avoid spamming.

VI. RELATED WORK

Challenges in requirements and testing alignment are investigated by many researchers. For instance, our previous work [3] assesses challenges with writing and updating requirements and acceptance tests. Bjarnason et al. [8] identified the challenges in maintaining the alignment of requirements and verification and validation activities when requirements change. Their results are confirmed by Larsson and Borg [9]. The results from the exploratory part of this study confirm that the challenges identified in [3] and [8] are still present. In this study we, not only identify the current challenges with requirements and test alignment, but also investigate which of them GuideGen can mitigate.

In order to facilitate the requirements change management process and to establish automated traceability for change impact analysis, researchers propose information retrieval (IR) [20], [21], [22] and natural language processing (NLP) [23], [24] methods. However, all these methods investigate how changes in requirements impact code or other requirements; they do not consider acceptance tests. Moreover, they only identify which documents are affected, but not how they should be changed, despite the recognized need for suggestions about how to handle requirements changes [25]. GuideGen provides concrete suggestions on how to adapt acceptance tests based on the analyzed changes in requirements.

Many researchers (e.g. [26], [27], [28], [29]) studied the usage of acceptance tests for specifying requirements. This is particularly applied in agile projects, in which acceptance criteria are used for specifying details of requirements. However, replacing requirements by acceptance criteria is not widely accepted in practice. This study also confirms that in many companies today, requirements documents are written, either as user stories or in traditional textual form.

In order to align requirements engineering and testing activities, researchers focused on defining formal models from informal requirements and automatically generating tests from these formal models [30], [31], [32], [33]. For instance, Escalona et al. [31] provide an overview of the research with regard to generating tests from functional requirements. They concluded that in order to have a successful test generation,

requirements would need to be specified in a much more formal way than they usually are in practice. Similarly, Post et al. [32] define a method for linking functional requirements to tests by creating a formal specification of requirements. Since practitioners are not keen in writing “code-like” or fully structured requirements, GuideGen aims at providing an approach that does not require practitioners to adapt the level of formalism in their requirements.

VII. CONCLUSION

In this paper, we report on a qualitative assessment of the usefulness and practical applicability of GuideGen, a method and tool for keeping requirements and acceptance test documents aligned via automatically generated guidance. The study is based on the data obtained from thirteen interviews in ten companies with twenty-three industrial practitioners. Based on the practitioners’ opinions, we assessed how useful GuideGen is for meeting challenges with evolving requirements and how applicable it is for the current setup in the companies.

The results show that GuideGen is useful for improving (1) communication in cross-functional and distributed teams, and (2) the process of updating acceptance tests when requirements change. Having suggestions on how to adapt the acceptance tests based on the changes in requirements is evaluated as superior to only being informed about the changes in a requirement. Communicating changes via automatically generated e-mails and warnings is seen as a positive aspect of the approach, although some practitioners expressed concerns about a potential flood of notifications when changes occur frequently. In addition, some practitioners prefer verbal communication, but they see these features as a good reminder to adapt the affected tests.

In summary, our study confirms that the core ideas of GuideGen are useful in industrial practice. Concerning the current state of the GuideGen tool, the study shows that the tool is applicable for smaller projects as it is. For bigger projects, however, GuideGen would either have to be incorporated as a plug-in into existing document and project management tools, such as Jira or TFS, or it would have to be extended with numerous features in order to make it a useful and practical stand-alone tool.

In our future work, we plan to investigate how GuideGen can be implemented as a plug-in for some of the tools currently used in practice. Further, we plan to address the limitation that GuideGen only allows one acceptance test per requirement. Supporting one-to-many relationships between requirements and acceptance tests introduces a new challenge: we will have to develop a method for identifying which acceptance tests are impacted by a change in the related requirement.

ACKNOWLEDGEMENTS

We thank the participants and their companies for their time and effort which made this study possible. This work was partially funded by the Swiss National Science Foundation under grant 200021-157004/1.

REFERENCES

- [1] G. J. Myers, C. Sandler, and T. Badgett, *The Art of Software Testing*. John Wiley & Sons, 2011.
- [2] E. J. Usitalo, M. Komssi, M. Kauppinen, and A. M. Davis, "Linking requirements and testing in practice," in *16th IEEE International Requirements Engineering Conference (RE'08)*. IEEE, 2008, pp. 265–270.
- [3] S. Hotomski, E. Ben Charrada, and M. Glinz, "An exploratory study on handling requirements and acceptance test documentation in industry," in *24th IEEE International Requirements Engineering Conference (RE'16)*. IEEE, 2016, pp. 116–125.
- [4] H. Ahmed, A. Hussain, and F. Baharom, "Current challenges of requirement change management," *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 8, no. 10, pp. 173–176, 2016.
- [5] P. Karhapää, A. Haghighatkhah, and M. Oivo, "What do we know about alignment of requirements engineering and software testing?" in *21st International Conference on Evaluation and Assessment in Software Engineering*. ACM, 2017, pp. 354–363.
- [6] S. Hotomski, E. Ben Charrada, and M. Glinz, "Keeping evolving requirements and acceptance tests aligned with automatically generated guidance," in *24th International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ 2018)*. Springer, 2018, pp. 247–264.
- [7] B. A. Kitchenham, "Evaluating software engineering methods and tool. part 1: The evaluation context and evaluation methods," *ACM SIGSOFT Software Engineering Notes*, vol. 21, no. 1, pp. 11–15, 1996.
- [8] E. Bjarnason, P. Runeson, M. Borg, M. Unterkalmsteiner, E. Engström, B. Regnell, G. Sabaliauskaite, A. Loconsole, T. Gorschek, and R. Feldt, "Challenges and practices in aligning requirements with verification and validation: a case study of six companies," *Empirical Software Engineering*, vol. 19, no. 6, pp. 1809–1855, 2014.
- [9] J. Larsson and M. Borg, "Revisiting the Challenges in Aligning RE and V&V: Experiences from the Public Sector," in *1st International Workshop on Requirements Engineering and Testing (RET 2014)*. IEEE, 2014, pp. 4–11.
- [10] S. Hotomski and M. Glinz, "GuideGen: a tool for keeping requirements and acceptance tests aligned," in *Proceedings of the 40th International Conference on Software Engineering (ICSE 2018), Companion volume*. ACM, 2018, pp. 49–52.
- [11] M. Q. Patton, *Qualitative Evaluation and Research Methods*. SAGE Publications, 1990.
- [12] G. Adzic, *Bridging the Communication Gap*. London: Neuri Ltd., 2009.
- [13] E. Ben Charrada, A. Koziolk, and M. Glinz, "Identifying outdated requirements based on source code changes," in *20th IEEE International Requirements Engineering Conference (RE'12)*. IEEE, 2012, pp. 61–70.
- [14] M. Daun, T. Weyer, and K. Pohl, "Detecting and correcting outdated requirements in function-centered engineering of embedded systems," in *21st International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ 2015)*. Springer, 2015, pp. 65–80.
- [15] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical Software Engineering*, vol. 14, no. 2, p. 131, Dec 2008.
- [16] B. Ryan, R. W. Scapens, and M. Theobald, *Research Method and Methodology in Finance and Accounting*. London: Thomson, 2002.
- [17] E.-M. Ihantola and L.-A. Kihn, "Threats to validity and reliability in mixed methods accounting research," *Qualitative Research in Accounting & Management*, vol. 8, no. 1, pp. 39–58, 2011.
- [18] I. Bolic. Gherkin syntax. [Online]. Available: <https://github.com/cucumber/cucumber/wiki/Gherkin>
- [19] F. D. Davis, "Perceived usefulness, perceived ease of use, and user acceptance of information technology," *MIS Quarterly*, vol. 13, no. 3, pp. 319–340, 1989.
- [20] A. De Lucia, A. Marcus, R. Oliveto, and D. Poshyanyk, "Information retrieval methods for automated traceability recovery," in *Software and Systems Traceability*. Springer, 2012, pp. 71–98.
- [21] A. Marcus, J. I. Maletic, and A. Sergeyev, "Recovery of traceability links between software documentation and source code," *International Journal of Software Engineering and Knowledge Engineering*, vol. 15, no. 05, pp. 811–836, 2005.
- [22] H. Eyal-Salman, A.-D. Seriai, and C. Dony, "Feature-to-code traceability in a collection of software variants: Combining formal concept analysis and information retrieval," in *14th International Conference on Information Reuse and Integration (IRI 2013)*. IEEE, 2013, pp. 209–216.
- [23] C. Arora, M. Sabetzadeh, A. Goknil, L. C. Briand, and F. Zimmer, "Change impact analysis for natural language requirements: An NLP approach," in *23rd International Requirements Engineering Conference (RE'15)*. IEEE, 2015, pp. 6–15.
- [24] W. Chen, H. Chen, C. Tao, and C. Fei, "FACIA: A Fully Automatic Change Impact Analysis Method for Large Scale Requirements," in *2017 International Conference on Applied Mechanics and Mechanical Automation (AMMA 2017)*, 2017, pp. 419–427.
- [25] S. Nair, J. L. de la Vara, and S. Sen, "A Review of Traceability Research at the Requirements Engineering Conference (RE@21)," in *21st IEEE International Requirements Engineering Conference (RE'13)*. IEEE, 2013, pp. 222–229.
- [26] E. Bjarnason, M. Unterkalmsteiner, E. Engström, and M. Borg, "An industrial case study on test cases as requirements," in *16th International Conference on Agile Processes in Software Engineering and Extreme Programming (XP 2015)*. Springer, 2015, pp. 27–39.
- [27] F. Ricca, M. Torchiano, M. Ceccato, and P. Tonella, "Talking tests: an empirical assessment of the role of fit acceptance tests in clarifying requirements," in *9th International Workshop on Principles of Software Evolution (IWPSE 2007)*. ACM, 2007, pp. 51–58.
- [28] R. C. Martin and G. Melnik, "Tests and Requirements, Requirements and Tests: A Möbius Strip," *IEEE Software*, vol. 25, no. 1, pp. 54–59, 2008.
- [29] B. Haugset and T. Stalhane, "Automated acceptance testing as an agile requirements engineering practice," in *45th Hawaii International Conference on System Science (HICSS 2012)*, 2012, pp. 5289–5298.
- [30] Z. A. Barmi, A. H. Ebrahimi, and R. Feldt, "Alignment of requirements specification and testing: A systematic mapping study," in *Workshop Proceedings, 4th IEEE International Conference on Software Testing, Verification and Validation (ICST 2011)*. IEEE, 2011, pp. 476–485.
- [31] M. Escalona, J. J. Gutierrez, M. Mejías, G. Aragón, I. Ramos, J. Torres, and F. Domínguez, "An overview on test generation from functional requirements," *Journal of Systems and Software*, vol. 84, no. 8, pp. 1379–1393, 2011.
- [32] H. Post, C. Sinz, F. Merz, T. Gorges, and T. Kropf, "Linking functional requirements and software verification," in *17th IEEE International Requirements Engineering Conference (RE'09)*. IEEE, 2009, pp. 295–302.
- [33] A. Goel and A. Roychoudhury, "Synthesis and traceability of scenario-based executable models," in *Second International Symposium on Leveraging Applications of Formal Methods, Verification and Validation (ISoLA 2006)*. IEEE, 2006, pp. 347–354.